Student Name: **Stud id:** **Sect#:** **#:**

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages **QUIZ#3: Chapter 16_LP**

****

1) Name 2 kinds of Prolog statements: *fact* and *rule* .

2) In Prolog, the unification of terms is performed by = operator and the calculation of terms is done by *is* operator.

3) The left side of a Prolog rule is called *consequence* and the right side is called *antecedent* .

4) The Prolog query: `?- X is 20 mod 7, Y is 20*0.6.` produces:

   X= 6 and *Y= 12.0*

```
formT(X,Y,R):- X>=Y, R is X+Y.
formT(X,Y,R):- X<Y, R is X-Y.
```

5) The Prolog query: `formT(10,25,U).` produces: *U = -15*

```
rat([],[]).
rat([H],[H]).
rat([H1,H2|T],[H1,H2|TT]) :- H1 =< H2, rat(T,TT).
rat([H1,H2|T],[H2,H1|TT]) :- H1 > H2, rat(T,TT).
```

6) The Prolog query: `rat([15,8,11,7,20],X).` produces: *X = [8,15,7,11,20].*

7) "*X and Y are brothers if they have the same mother and the same father*" .The Prolog rule that describes this relation is:

   brothers(X,Y):- father(F,X) ,father(F,Y) ,mother(M,X) ,mother(M,Y) .

8) Write Prolog code named *quz* that accepts a list of numbers and produces a list each element of which is the product of the neighboring elements. If the list contains one element, then it squares it.

```
?- quz([2,8,4,5],U).
   U = [16, 20].
?- quz([2,5,4],U).
   U = [10, 16] .
```

```
quz([],[]).
quz([H],[R]):- R is H*H.
quz([H1,H2|T],[HR|TT]):- HR is H1*H2, quz(T,TT).
```

Student Name: **Stud id:** **Sect#:** **#:**

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages **QUIZ#3: Chapter 16_LP**

****

1) The Prolog operator =\= is similar to != operator in C++. The Prolog operator =:= is similar to== operator in C++.

2) A Prolog statement consists of *terms* and terminates by a *dot*.

3) The process of finding useful values for variables in propositions that allows matching process to succeed is called *unification*. The process of assigning temporary values to variables to allow unification to succeed is called *instantiation*.

4) The Prolog query: `?- X is 2**6, Y is 47 mod 5.` produces:

   X= 64 and *Y= 2*

```
taz([],[]).
taz([H1,H2|T],[HR|TT]):- HR is H2-H1, taz(T,TT).
```

5) The Prolog query: `taz([5,8,11,7,20,9],F).` produces: *F = [3, -4, -11].*

```
fat([],[]).
fat([H],[H]).
fat([H1,H2|T],[H1,H2|TT]) :- H1 >= H2, fat(T,TT).
fat([H1,H2|T],[H2,H1|TT]) :- H1 < H2, fat(T,TT).
```

6) The Prolog query: `fat([15,8,7,11,55],X).` produces: *X = [15,8,11,7,55].*

7) "*X and Y are brothers if they have the same mother and the same father*" .The Prolog rule that describes this relation is:

   brothers(X,Y):- father(F,X),father(F,Y),mother(M,X),mother(M,Y).

8) Write Prolog code named *zuq* that accepts a list of numbers and produces a list each element of which is the modulus of the two neighboring elements. If the list contains one element, then it halves it.

```
?- zuq([15,8,14,5],U).
   U = [7, 4].
?- zuq([20,6,24],U).
   U = [2, 12] .

zuq([],[]).
zuq([H],[R]):- R is H/2.
zuq([H1,H2|T],[HR|TT]):- HR is H1 mod H2, zuq(T,TT).
```